

Έστω A ένας μονοδιάστατος πίνακας N θετικών ακεραίων αριθμών που βρίσκονται στις θέσεις $A[0]$ έως $A[N-1]$. Να γράψετε μία κομψή και αποδοτική συνάρτηση $\text{count}(N, A)$ που να επιστρέφει το πλήθος των στοιχείων του πίνακα A που είναι μικρότερα ή ίσα από όλα τα επόμενά τους.

Παράδειγμα 1: ($N = 7$)

$A = [5, 4, 7, 6, 8, 6, 9]$

$\text{count}(N, A) = 4$

Παράδειγμα 2: ($N = 5$)

$A = [5, 4, 3, 2, 1]$

$\text{count}(N, A) = 1$ 🗨️

Εισαγάγετε την απάντησή σας

Γράψτε ένα κομψό και αποδοτικό πρόγραμμα που να διαβάζει ένα αρχείο κειμένου με όνομα "file.txt", αποτελούμενο από πεζά και κεφαλαία λατινικά γράμματα, αριθμητικά ψηφία (0, 1, ..., 9), κενά διαστήματα, και αλλαγές γραμμής. Το πρόγραμμά σας πρέπει να εκτυπώνει στην οθόνη το πλήθος των αριθμών που εμφανίζονται στο κείμενο. Ως αριθμός νοείται οποιαδήποτε συμβολοσειρά αποτελείται μόνο από ψηφία και χωρίζεται με κενά διαστήματα (ή την αρχή, ή το τέλος γραμμής) από το υπόλοιπο κείμενο.

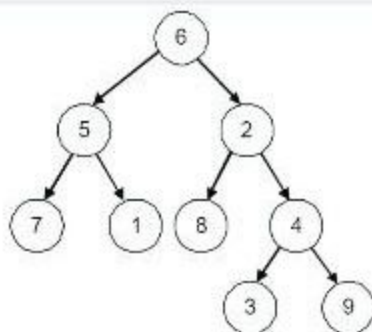
Παράδειγμα:


(αρχικό κείμενο):

In Greek NTUA is called the Ethnicon Metsovion Polytechnion to honor the donors and benefactors Nikolaos Stournaris Eleni Tositsa Michail Tositsas and Georgios Averof all originally from Metsovo who made substantial donations in the last half of the 19th century A rearrangement took place between 1862 and 1864 when Polytechnion was reorganized and expanded with the addition of more technical courses This tendency was continued in the period starting in 1864 and ending in 1873

(οθόνη):

4 



Ποια από τις παρακάτω προτάσεις είναι ψευδής για το δέντρο του σχήματος; 

- ☐ Καμία από τις άλλες προτάσεις δεν είναι ψευδής
- ☐ Η σειρά προθεματικής (prefix) διάσχισης είναι: 6 5 2 7 1 8 4 3 9
- ☐ δεν απαντώ
- ☐ Η σειρά ενθεματικής (infix) διάσχισης είναι: 7 5 1 6 8 2 3 4 9
- ☐ Η σειρά επιθεματικής (postfix) διάσχισης είναι: 7 1 5 8 3 9 4 2 6

Ποια είναι η τιμή της μεταβλητής p στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος;

```
int n=1, p=1;  
do { n*=2; p+=n; } while (n <= 1024);
```

- ☐ κανένα από τα αναφερόμενα
- ☐ 2047
- ☐ 2048
- ☐ 4095
- ☐ δεν απαντώ

Ποια είναι η τιμή της μεταβλητής `t` στο τέλος της εκτέλεσης του ακόλουθου τμήματος προγράμματος;

```
int n = 8192, t = 0;  
for (int i = 1; i <= n; i *= 2) { t++; n--; }
```

☐ κανένα από τα αναφερόμενα

☐ 12

☐ δεν απαντώ

☐ 13

☐ 11

Έστω ότι έχετε τρεις διαφορετικούς αλγορίθμους A, B και Γ, που επιλύουν το ίδιο πρόβλημα. Η πολυπλοκότητα του A είναι $O(n^{3/2})$, του B είναι $O(n (\log n)^6)$, και του Γ είναι $O((3/2)^n)$. Ποιον από τους τρεις θα προτιμούσατε; (Θεωρήστε ότι μας ενδιαφέρουν μεγάλες τιμές του n.) 🗣️

- ☐ δεν απαντώ
- ☐ οποιονδήποτε από τους A ή Γ, δεν έχουν διαφορά
- ☐ τον B
- ☐ τον Γ
- ☐ τον A

Τι επιστρέφει η παρακάτω συνάρτηση αν κληθεί με $x = 100$ και $n = 1024$;


```
int fun(int x, int n) {  
    if (n == 0) return x;  
    else return x + fun(x-1, n/2);  
}
```

- ☐ 1134
- ☐ 1045
- ☐ κανένα από τα αναφερόμενα
- ☐ 4950
- ☐ δεν απαντώ

Με βάση τον ορισμό δυαδικού δέντρου tree γράφτηκε η συνάρτηση foo. Τι υπολογίζει;

```
struct node {  
    int info;  
    node *left, *right;  
};  
  
typedef node *tree;  
  
int foo(tree x) {  
    if (!x) return 0;  
    else if (!x->left)  
        if (!x->right) return 0;  
        else return x->info + foo(x->right);  
    else return foo(x->left) + foo(x->right);  
}
```

- ☐ δεν απαντώ
- ☐ το άθροισμα των τιμών των εσωτερικών κόμβων που έχουν μόνο αριστερό υποδέντρο
- ☐ το άθροισμα των τιμών των εσωτερικών κόμβων
- ☐ το άθροισμα των τιμών των φύλλων
- ☐ το άθροισμα των τιμών των εσωτερικών κόμβων που έχουν μόνο δεξιό υποδέντρο

Τι τυπώνει το παρακάτω πρόγραμμα; 

```
PROGRAM {  
    int *p = new int, *q = new int, *t = new int;  
    *p=42; *q=2; *t=17;  
    p=t; t=q;  
    *t=*p**q ; *q=*q+*t;  
    WRITELN(*t);  
}
```

δεν απαντώ

κανένα από τα αναφερόμενα

42

68

252

Ποια από τις παρακάτω εντολές δεν προκαλεί την αύξηση κατά ένα του περιεχομένου της μεταβλητής x ;

- ☐ $x+=1;$
- ☐ $x+1;$
- ☐ $x++;$
- ☐ δεν απαντώ
- ☐ $++x;$

Από το παρακάτω τμήμα κώδικα λείπει λίγος κώδικας, στη θέση που σημειώνεται με το σχόλιο. Τι πρέπει να συμπληρωθεί προκειμένου η συνάρτηση `median`, αν της δοθεί μία ταξινομημένη λίστα ακεραίων αριθμών, να επιστρέφει το διάμεσο των τιμών της λίστας; (Υποθέστε ότι το πλήθος των στοιχείων της λίστας θα είναι περιττός αριθμός.) 📄

```
struct node {
    int info;
    node *next;
};

typedef node *list;

int median(list x) {
    list m = x;
    while (x) {
        x = x->next;
        if (!x) break;
        /* εδώ λείπει λίγος κώδικας */
    }
    return m->info;
}
```


- ☐ `x = x->next; m = m->next;`
- ☐ δεν απαντώ
- ☐ `x = x->next;`
- ☐ `m = m->next;`
- ☐ `m = x; x = x->next;`

Ποιο από τα παρακάτω προγράμματα τυπώνει 42; 

```
int k=4;
PROC proc1(int &n) {
    n=k*(k+1); WRITELN(n/2*k+2);
}
PROGRAM { proc1(k); }
```

```
int k=4;
PROC proc2(int n) {
    n=k*(k+1); WRITELN(n/2*k+2);
}
PROGRAM { proc2(k); }
```

- ☐ το δεξιό
- ☐ και τα δύο
- ☐ το αριστερό
- ☐ δεν απαντώ
- ☐ κανένα από τα δύο

Τι τυπώνει το παρακάτω πρόγραμμα; 

```
int p=3, q=4;  
PROC proc(int &x, int y) {  
    x=x*y; y=x+p;  
    WRITELN (x+y+p+q) ;  
}  
PROGRAM { proc(p, q) ; }
```


☐ κανένα από τα αναφερόμενα

☐ 35

☐ 52

☐ δεν απαντώ

☐ 34

Τι επιστρέφει η συνάρτηση `search` αν κληθεί με `low = -10` και `up = 100`; 

```
int f(int x) { return x*x-10*x-200; }

int search(int low, int up) {
    int mid=(low+up)/2, val=f(mid);
    if (val == 0) return mid;
    if (val*f(up) < 0) return search(mid+1, up);
    else return search(low, mid-1);
}
```

- ☐ 20
- ☐ 5
- ☐ δεν απαντώ
- ☐ κανένα από τα αναφερόμενα
- ☐ -10

Από το παρακάτω τμήμα κώδικα λείπει μία εντολή, στη θέση που σημειώνεται με το σχόλιο.

```
int p=1;  
for (int i=2; i<=3; i++) /* εδώ λείπει μία εντολή */  
cout << p*(p+1) << endl;
```

Ποια από τις παρακάτω εντολές πρέπει να συμπληρωθεί ώστε να εκτυπώνεται ο αριθμός 42;

A. `p += i;`

B. `p *= i;`

- ☐ καμία από τις δύο
- ☐ οποιαδήποτε από τις δύο
- ☐ δεν απαντώ
- ☐ η B αλλά όχι η A
- ☐ η A αλλά όχι η B

Τοποθετήστε τους παρακάτω αλγορίθμους σε αύξουσα σειρά (από την μικρότερη πολυπλοκότητα στη μεγαλύτερη) ως προς την χρονική πολυπλοκότητά τους στη χειρότερη περίπτωση:

- A. ταξινόμηση με συγχώνευση (mergesort) ενός πίνακα n στοιχείων
- B. δυαδική αναζήτηση (binary search) σε ταξινομημένο πίνακα n στοιχείων
- Γ. ταξινόμηση φουσαλίδας (bubble sort) ενός πίνακα n στοιχείων

☐ Γ, Β, Α

☐ δεν απαντώ

☐ Α, Γ, Β

☐ Β, Γ, Α

☐ Β, Α, Γ

Τι τυπώνει το παρακάτω πρόγραμμα;

```
PROGRAM {  
    int *p = new int[10], *q;  
    int sum=0, i;  
    for (i=1; i<=10; i++) *(p+i-1)=i;  
    for (i=2, q=p+2; i<10; i++, q++) sum+=*(q-1)+*(q-2);  
    WRITELN(sum);  
}
```

- ☐ 55
- ☐ 110
- ☐ κανένα από τα αναφερόμενα
- ☐ δεν απαντώ
- ☐ 80